

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2000-330836  
(P2000-330836A)

(43) 公開日 平成12年11月30日 (2000.11.30)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テームコード (参考)
G 0 6 F 12/00	5 1 2	G 0 6 F 12/00	5 1 2 5 B 0 8 2

審査請求 有 請求項の数4 O L (全 11 頁)

(21) 出願番号 特願平11-140931

(22) 出願日 平成11年5月21日 (1999.5.21)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 大野 秀俊

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100093838

弁理士 小橋川 洋二

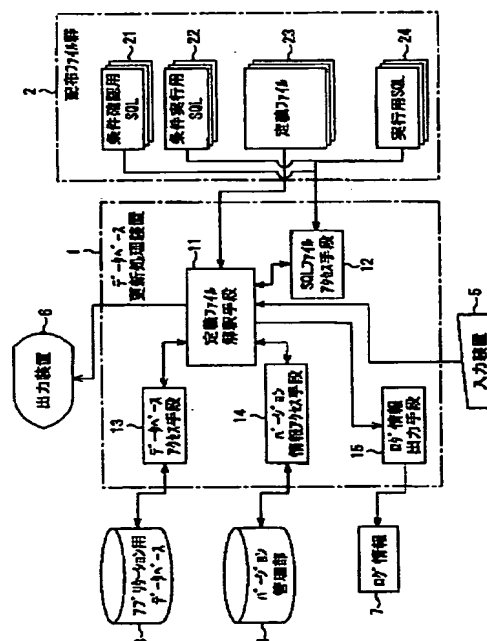
Fターム (参考) 5B082 GA07 GA08

(54) 【発明の名称】 構築済みデータベースの変更方式

(57) 【要約】

【課題】 アプリケーションソフトウェアによって既にデータベース上に構築されている表の設定やデータを、別の実行プログラム配布すること無しに、変更可能にした構築済みデータベースの変更方式を提供する。

【解決手段】 アプリケーションソフトウェアによって構築された情報を有するデータベース3における前記情報の変更が必要の場合に、所定条件を備えた文字情報からなるファイル群2を配布し、前記データベースの情報を更新するようにした。



## 【特許請求の範囲】

【請求項1】 アプリケーションソフトウェアによって構築された情報を有するデータベースにおける前記情報の変更が必要の場合に、所定条件を備えた文字情報からなるファイル群を配布し、前記データベースの情報を更新するようにしたことを特徴とする構築済みデータベースの変更方式。

【請求項2】 前記配布されるファイル群は、条件確認用SQLと、条件実行用SQLと、定義ファイルと、実行用SQLとを備えてなることを特徴とする請求項1記載の構築済みデータベースの変更方式。

【請求項3】 前記データベースにおける更新した情報のバージョンを管理することを特徴とする請求項1または請求項2記載の構築済みデータベースの変更方式。

【請求項4】 前記文字情報の配布は、ダウンロードにより行うことを特徴とする請求項1乃至請求項3のいずれかに記載の構築済みデータベースの変更方式。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、構築済みデータベースの変更方式に関し、特にアプリケーションソフトウェアによって既にデータベース上に構築されている表の設定やデータを変更する場合に、別の実行プログラムを配布しないでも、前記変更をすることが可能な構築済みデータベースの変更方式に関する。

## 【0002】

【従来の技術】従来は、アプリケーションにより既に構築されているデータベースの内容（表やデータ等の情報）を変更する場合に、データベースの内容を変更する為の実行プログラムを配布し実行していた。

## 【0003】

【発明が解決しようとする課題】しかしながら、この従来の技術には、次のような問題点があった。第1の問題点は、前記実行プログラムの配布後に発生した変更を含めて、統合して管理できないということである。その理由は、データベースを更新するプログラムは、既存のデータベースを変更するために配布されるものなので、アプリケーション側で後に変更されるデータベースのバージョンの管理を一元管理する機能を持たないためである。

【0004】第2の問題点は、データベースの変更の場合であっても、配布プログラムをセットアップ（インストール）する必要があるということである。その理由は、プログラムは一般にセットアップを行った後に利用するものであるためである。

【0005】第3の問題点は、配布プログラムは、文字情報で構成されたファイルと比較した場合に、通信（例えば、ダウンロード）を用いた配布に適さない（多大な時間を要する）ということである。その理由は、文字情報であるファイル（例えば、数Kバイト）に比べ、プログラムは容量が大きいためである（例えば、200Kバ

イト）。

【0006】第4の問題点は、配布プログラムは、簡単に作成できないということである。その理由は、プログラムは一般にソースコードを記述し、コンパイルを行った後にセットアップを作成する必要があるためである。

【0007】そこで本発明の課題は、アプリケーションソフトウェアによって既にデータベース上に構築されている表の設定やデータを、別の実行プログラム配布すること無しに、変更可能にした構築済みデータベースの変更方式を提供することである。

## 【0008】

【課題を解決するための手段】前記課題を解決するために本発明は、アプリケーションソフトウェアによって構築した情報を有するデータベースと、該データベースの前記情報の変更が必要になった際に配布される所定条件を備えた文字情報からなるファイル群と、該ファイル群の所定条件を満足する場合に、前記データベースの情報を更新するデータベース更新手段とを備えたことを特徴とする。

【0009】このようにすれば、データベースの情報の変更が必要な際に配布されるファイル群は文字情報なので、例えばダウンロードする際の情報量が少なく済むと共にシステムエンジニアではなく一般ユーザが更新作業を行うことができ、システムエンジニアがバージョンチェック等の作業をする必要がないので、この作業の際の所謂ボカミス（ケアレスミス）を防止することができる。

## 【0010】

【発明の実施の形態】以下、本発明を図示の実施例に基づいて説明する。

## 【0011】[1] 第1実施例

## (1) 構成

図1を参照すると、本実施例は、アプリケーションソフトウェアに含まれプログラム制御により動作するデータベース更新処理装置1と、必要に応じてユーザに配布（例えば、ダウンロードにより）される「文字情報」からなる配布ファイル群2と、アプリケーションによって構築済みのアプリケーション用データベース3と、アプリケーションによって構築済みのバージョン管理部4と、キーボード等の入力装置5と、ディスプレイ装置等の出力装置6と、作業内容を記録するログ情報7とを含む。

## 【0012】①データベース更新処理装置1

データベース更新処理装置1は、定義ファイル解釈手段11と、SQL(Structured Query Language)ファイルアクセス手段12と、データベースアクセス手段13と、バージョン情報アクセス手段14と、ログ情報出力手段15とを備える。

【0013】定義ファイル解釈手段11は定義ファイル23に逐次記述されている内容を元に、アプリケーション

10

20

30

40

50

ン用データベース3を更新する。更新の際には次に説明する条件確認用SQL21、条件実行用SQL22、実行用SQL24から更新内容の情報を取得する。また、定義ファイル23に記述されている内容（例えば、更新情報等）を元に、バージョンの確認や変更を行う。実行結果はログ情報7に出力される。

【0014】SQLファイルアクセス手段12は、次に説明する条件確認用SQL21と条件実行用SQL22、および実行用SQL24の内容を定義ファイル解釈手段11の指示に基づき読み出す。

【0015】データベースアクセス手段13は、定義ファイル解釈手段11の指示に基づきアプリケーション用データベース3を参照・更新する。これは使用するデータベースにより、データベースが用意するミドルウェアも含まれる。バージョン情報アクセス手段14は、定義ファイル解釈手段11の指示に基づきバージョン管理部4を参照・更新する。ログ情報出力手段15は、定義ファイル解釈手段11の実行内容をログ情報7に出力する。

【0016】②配布ファイル群2

配布ファイル群2は、前述の条件確認用SQL21と、条件実行用SQL22と、定義ファイル23と、実行用SQL24とを備える。条件確認用SQL21は、条件実行用SQL22を実行する条件となる戻り値をアプリケーション用データベース3に問合せるSQLを記述する。

【0017】条件実行用SQL22は、定義ファイル解釈手段11が条件確認用SQL21を使用してアプリケーション用データベース3より取得した値と定義ファイル23に記述されている内容と比較し、条件が合致した場合にアプリケーション用データベース3に対して実行するSQLを記述する。

【0018】定義ファイル23は、定義ファイルヘッダ部と定義ファイル実行部で構成される。定義ファイルヘッダ部には、その定義ファイル名とバージョン情報、実行可能なバージョンの条件等が記述される。定義ファイル実行部には、アプリケーション用データベースを更新する内容やバージョンを参照・更新する内容、およびユーザに対する問合せの情報を逐次記述する。実行用SQL24は、アプリケーション用データベース3に対して実行するSQLを記述する。

【0019】③アプリケーション用データベース3

アプリケーション用データベース3は、アプリケーションにより既に構築されているデータベースであり、配布ファイル群2の更新対象である。

【0020】④バージョン管理部4は、アプリケーションソフトウェアにより構築されており、バージョン情報を格納する。バージョン管理部4の所在の例として、アプリケーション用データベースに含まれる場合と、レジストリデータベースに含まれる場合と、ファイルに記録

されている場合等がある。

【0021】⑤ログ情報7は、定義ファイル解釈手段11の実行内容を記録する。ログ情報の所在の例として、任意のファイルや、アプリケーション用データベース3等がある。

【0022】(2)動作

次に、図1～図13を参照して本実施例の全体の動作について詳細に説明する。

【0023】①全体の概略動作

10 図2は本実施例の全体動作を示す概略フローチャートである。データベース更新処理装置1が起動されると、図2の処理が定義ファイル解釈手段11で実行される。即ち、定義ファイル解釈手段11は、アプリケーション用データベース3に接続する。接続するユーザの設定方法はアプリケーションにより規定される（ステップA01）。次に、実行可能な定義ファイルの所在を確認する（ステップA02）。規定値の位置に定義ファイルが存在しない場合、利用者にその所在を問合せる。

【0024】定義ファイル23が発見された場合は、その定義ファイルヘッダ部より定義ファイル23の名称と動作可能なバージョン情報を取得する（ステップA03）。ここで定義ファイル23のヘッダ部に必要なSQLファイル一覧も記述し、使用するファイルの存在チェックを行ってもよい。定義ファイル解釈手段11はバージョン情報アクセス手段14を利用して、バージョン管理部4の情報を読み出す。定義ファイル23より読み出したバージョン情報とバージョン管理部4から読み出した情報が合致した場合、実行可能として出力装置6より利用者に表示する（ステップA04、および、ステップA05）。

【0025】実行不可の場合でも、実行不可として利用者に表示したほうがよい。利用者は表示された一覧の中より実行する定義ファイルを選択する（ステップA06）。定義ファイル解釈手段11は、選択された定義ファイルの実行部を逐次読み出し、実行する（ステップA07）。読み出し可能な定義ファイルの実行部がなくなると、処理は終了する。

【0026】②図3は、定義ファイル23の実行部の読み出しである。図3において、ステップB01で、定義ファイルの実行部の1行を読み出す。ステップB02にて、読み出せたか確認する。読み出せた場合は、ステップB03で、実行可能行であるか確認する。本実施例の場合、実行可能でない行とは、空白行、コメント行、ラベル行、不正な行を指す。ステップB04で、読み出した内容を実行し、ステップB01へ戻る。ステップB02にて読み出せない場合は、終了する。

【0027】③図4は、読み出した定義ファイル23の行がバージョンの確認である場合の処理である。ステップC01で、ステップB03にて読み出された情報より、バージョン情報の名称と、その条件、および、分岐

先のラベル情報を取得する。ステップC02で、バージョン情報アクセス手段14を利用し、ステップC01で読み出されたバージョン情報の名称よりバージョンを取得する。

【0028】ステップC03にて、定義ファイルのバージョンの条件とバージョン管理部のバージョン情報を比較する。本実施例では、比較は等号、不等号(=、<、>、>=、<=、!=)が使用され、バージョン情報が文字列の場合は、その文字コードにて比較を行う。条件が合致した場合は、ステップC04にて定義ファイルの読み出し位置をステップC01にて読み出したラベル情報の位置に移動し終了する。合致しない場合は、そのまま終了する。

【0029】④図5は、ステップB03にて読み出した定義ファイル23の行がバージョンの変更である場合の処理である。ステップD01で、ステップB03にて読み出された情報より、バージョン管理部4の追加・変更の内容を取得する。ステップD02にて、バージョン情報アクセス手段14を介し、バージョン情報を追加・修正する。

【0030】⑤図6は、ステップB03にて読み出した定義ファイルの行が、条件で実行されるSQLである場合の処理である。ステップE01で、ステップB03にて読み出された情報より、条件確認用SQL21のファイル名と、その条件と、条件実行用SQL22のファイル名を取得する。ステップE02で、SQLファイルアクセス手段12を介し、条件確認用SQL21の内容を読み出す。

【0031】ステップE03にて、データベースアクセス手段13を介し、条件確認用SQL21を実行し、その結果を取得する。ステップE04にて、取得した結果とステップE01にて取得した条件とを比較する。比較が合致した場合は、ステップE05で、条件実行用SQL22を読み出し、アプリケーション用データベース3に対して実行し終了する。条件が合致しない場合は、終了する。

【0032】⑥図7は、ステップB03にて読み出した定義ファイルの行が無条件に実行されるSQLである場合の処理である。ステップF01で、ステップB03にて読み出された情報より、実行用SQL24のファイル名を取得する。ステップF02で、ステップF01で取得したファイル名で実行用SQL24の内容を取得する。次にステップF02にて、データベースアクセス手段13を介し、アプリケーション用データベース2に対してSQLを実行する。

【0033】⑦図8は、ステップB03にて読み出した定義ファイルの行が利用者への問合せである場合の処理である。ステップG01で、ステップB03にて読み出された情報より、メッセージの内容と、問合せ形式を取得する。ステップG02で、問合せ形式が分岐のもので

あれば、分岐先のラベル情報を取得する。ステップG03で、利用者に対しメッセージを表示し、問合せ形式に応じた入力部を出力装置6に表示する。利用者は内容を確認し、入力する。ステップG04で、入力された内容が肯定(「はい」「OK」)である場合、定義ファイルの読み出し位置をステップG02にて読み出したラベル情報の位置に移動し終了する。その他が選択された場合は、終了する。

【0034】⑧図9は、ステップB03にて読み出した定義ファイル23の行が定義ファイル読み出し行の移動である場合の処理である。ステップH01で、ステップB03にて読み出された情報より、移動先のラベル名称を取得する。ステップH02で定義ファイルの読み出し位置をステップH01にて読み出したラベル情報の位置に移動し終了する。

【0035】(3)次に、具体例を用いて説明する。例えば、図10に示すようなアプリケーション用データベースと、バージョン管理部と、定義ファイルと、条件確認用SQLと、条件実行用SQLと、実行用SQLが用意されたとする。定義ファイルは、一文字目が「#」であるものをコメント行と定義する。また、「:」で始まる行をラベル行と定義する。[HEADER]、[EXECUTE]は、処理部を宣言するものと定義する。その他の行は、「要素」で構成される行と定義し、要素は「,」で区切られているものとする。区切られた要素は、左から順に番号を割り当てる。

【0036】第一要素が「T」であるものを定義ファイルのタイトル行とし、第二要素をタイトルの文字列と定義する。第一要素が「C」であるものをバージョン確認情報とし、第二要素をシステム名、第三要素を比較詞、第四要素を比較情報と定義する。第一要素が「VC」であるものは、バージョンの確認の情報の行とし、第二要素をシステム名、第三要素を比較詞、第四要素を比較情報、第五要素を合致時の移動先ラベルと定義する。

【0037】第一要素が「V」の行はバージョンの記述情報の行であり、第二要素がシステム名、第三要素がバージョン情報と定義する。第一引数が「SC」である行は条件SQLの情報の行とし、第一要素が条件確認用SQLファイル名、第二要素が比較詞、第三要素が比較条件、第四要素が条件実行用SQLファイル名と定義する。

【0038】第一引数が「S」の行は、SQLの実行の情報が記述されており、第二要素が実行用SQLファイル名と定義する。第一要素が「Q」である場合は利用者への問合せに関する情報が記述されており、第二要素がメッセージ内容、第三要素が問合せ形式、第四要素が移動先ラベルと定義する。第三要素は「1」が「確認」形式、「2」が「はい」「いいえ」形式、「3」が「OK」「キャンセル」形式とし、「1」の場合は第四要素は必要無いものと定義する。第一要素が「J」の行は移

動に関する記述がされており、第二要素が移動先のラベルと定義する。

【0039】定義ファイル解釈手段11はステップA02で定義ファイルを確認し、ステップA03で定義ファイルのヘッダ部より定義ファイル名とバージョン確認情報を取得する。具体的には1行目([1]の行)より読み込み、[1]行の「#」で始まる行はコメント行として次の行を読み込む。次の[2]行で「[HEDDER]」を発見し、以下がヘッダ情報と認識する。次の[3]行を読み、第一要素が「T」で始まるものをタイトル情報と認識し記憶する。次の[4]行を読み、第一要素が「C」で始まるものをバージョン確認情報として読み込み記憶する。

【0040】同様に次の[5]行も読み込み、記憶する。次の行および[6]行は空行とコメント行のため読み飛ばす。次の[7]行で「[EXECUTE]」を読み込み、ヘッダ部が終了したことを認識する。次にステップA04にてバージョン管理部と比較する。

【0041】具体的には、一つ目のバージョン確認情報、「販売管理システム」のバージョンが、「1.1」「以上(=)」であるので、合致する。二つ目の「販売管理システムアドオン」はバージョン管理部に存在しないので、バージョン情報アクセス手段は空文字を返すが、文字比較を行う場合は空文字の方が小さくなるため、二つ目のバージョン確認情報も合致する。全てのバージョン確認情報が合致するので、ステップA05で利用者に実行可能な設定ファイルとして表示する。複数の設定ファイルがある場合は、同様に設定ファイルのヘッダ部を確認し、利用者に対する一覧中に表示する。ステップA06で、利用者は設定ファイルを選択し、実行する。

【0042】実行されると、ステップB01で定義ファイルを1行読み込む。具体的には、[8]行が読み込まれ、ステップB02、ステップB03で実行可能な行と認識する。ステップB04で第一要素がVCであるものはバージョンの確認であると認識する。従って、ステップC01から実行し、「販売管理システムアドオン」のバージョンが「1.0」「より小さい(<)」か確認する。ここでもバージョン情報アクセス手段は空文字を返すので、条件に合致するため、ステップC04で「MS」ラベルへ定義ファイルの読み込み位置を変更する。「:」で始まる行がラベルであるので、読み込み位置は「:MS」の[16]行に変更される。次の[17]行を読み込み、第一要素が「Q」であるため、ステップG01でメッセージと問合せ形式と取得する。

【0043】問合せ形式「2」が「はい」「いいえ」形式であるので、ステップG02で第四要素「V1」をラベル情報として読み込む。利用者に第二要素のメッセージおよび「はい」「いいえ」の入力部分を出力装置6に出力する。ステップG04にて利用者が「いいえ」を選択した場合は、ラベルへの移動は行われずに[18]行を

経て処理は終了する。利用者が「はい」を選択した場合は、ラベル情報に従い、定義ファイルの読み込み位置を[13]行に変更する(ステップG05)。

【0044】次に[14]行を読み込むと、第一要素が「SC」であるため条件SQLの実行を行う情報があると認識し、各要素を読み込む(ステップE01)。ステップE02で条件確認用SQL「CHECK.SQL」を読み込み(ステップE02)、SQLをアプリケーション用データベース3に対して実行する(ステップE03)。アプリケーション用データベース3はSQLを実行し、該当テーブルが無いため、「0」を返す。この結果と比較し、合致するので、第四要素の「C\_TABLE.SQL」が実行される(ステップE04、ステップE05)。実行結果を図11に示す。

【0045】次の[15]行を読み込むと、第一引数が「J」であるので、移動の情報が記述されていると認識する。第二要素を読み込み、(ステップH01)移動先ラベル「GO」より[9]行へ定義ファイルの読み出し位置を変更する。(ステップH02)次の[10]行を読み込むと、第一要素が「S」であるので、SQLの実行に関する情報が記述されている行と認識する。第二要素を読み込み(ステップF01)、実行用SQL「DATA.SQL」を読み込み実行する(ステップF02)。実行結果を図12に示す。

【0046】次の[11]行を読み込むと、第一要素が「V」であるのでバージョンの変更の行であると認識する。各要素を読み込み、(ステップD01)バージョン管理部の情報を変更する。(ステップD02)ここで、バージョン管理部には「販売システムアドオン」の情報が無いため、この情報は作成される。結果を図13に示す。

【0047】[15]行で読み込み位置が移動し、定義ファイルの読み込み対象が無くなるので処理を終了する(ステップB01、ステップB02)。ここで利用者には一覧が再度表示されるが、[5]行の条件に合致しなくなっているため、再度実行することはできない。

【0048】全ての処理の途中で更新に失敗した場合はデータベースをロールバックさせてもよい。ロールバックさせる場合は、処理開始時にセーブポイントが発生させたほうがよい。また、実行するSQLはロールバック可能なSQLに限定される。

#### 【0049】[II] 第2実施例

次に、本発明の第2実施例について図面を参照して詳細に説明する。

【0050】図14を参照すると、本実施例は、データベース更新処理装置8が、図1に示された実施例におけるデータベース更新処理装置1に加え、文字置換手段16とアプリケーション情報アクセス手段17を有する点と、アプリケーション情報9を有する点が異なる。

【0051】アプリケーション情報9は、レジストリ

データベース、設定ファイル、アプリケーション用データベース内の情報等、情報の取得場所は問わず、アプリケーションによって情報の所在を規定する。アプリケーション情報アクセス手段17は要求のあった置換文字列に従い、アプリケーション情報より置換後文字列を取得する。文字置換手段16は、配布ファイル群2より読み込まれた文字情報よりアプリケーションで規定されている置換文字列を検索し置換する。

【0052】次に図15のフローチャートを参照して、本実施例の動作について説明する。図15のフローチャートは図2～9のフローチャートの配布ファイル群が読み込み時に適用される。

【0053】定義ファイル解釈手段11より配布ファイル群2に読み取り要求が発生すると、文字置換手段16は配布ファイルの要求箇所を読み取る(ステップ101)。読み取られた文字列よりアプリケーションで規定される置換文字列を検索する(ステップ102)。置換文字列が発見されると(ステップ103)、アプリケーション情報アクセス手段17を経てアプリケーション情報より置換後文字列を取得する(ステップ104)。取得された置換後文字列と置換文字列を置換し(ステップ105)、置換文字列が発見されなくなるまで処理を繰り返す。置換文字列が全て変換されると、定義ファイル解釈手段11に結果を戻す。

【0054】次に、具体例について説明する。図16の具体例では、アプリケーションにより「%COM\_NAME%」と「%APP\_PATH%」が規定されている。

【0055】定義ファイル解釈手段11が定義ファイルより[3]行を実行する場合、図15のフローチャートが開始される。文字置換手段16は[3]行を読み込み、文字列「T,%COM\_NAME%販売システムアドオンVER1.1」を取得する(ステップ101)。取得された文字列より置換文字列を検索すると、「%COM\_NAME%」を発見する(ステップ102、ステップ103)。

【0056】発見された置換文字列よりアプリケーション情報アクセス手段17を経て置換後文字列「〇〇会社」を取得する(ステップ104)。取得文字列を置換すると、「T,〇〇会社販売システムアドオンVER1.1」の文字列が作成される(ステップ105)。作成された文字列を再検索すると、置換文字列を発見できないため、処理は終了し、定義ファイル解釈手段11に置換後の文字列が渡される。同様に[14]行を読み込むと、定義ファイル解釈手段11は、「SC,CHECK,SQL,=,0,C:%APP%CTABLE.SQL」を得る。

【0057】図14を参照すると、本実施例では複数のSQLファイルを使用し、全て同じデータベースユーザ、ステップA01で接続するユーザでSQLを実行す

るが、各SQLに実行ユーザを記述してもよい。記述箇所は設定ファイルの要素を増やすのが最もよい。またSQLファイルに記述してもよい。記述内容は、ステップA01で接続するユーザを指定する方法と、直接ユーザ名を記述する方法、共に指定できるのがよい。ステップA01で接続するユーザを使用する場合は、文字置換手段16に含め、置換文字列でユーザ名を置換してもよい。

【0058】

【発明の効果】以上説明したように本発明によれば、以下の効果を奏することができる。第1の効果は、バージョンの違いや適用するシステムの間違いによるトラブルを防ぐことができることにある。その理由は、適用するシステムとバージョンや配布ファイルのバージョンを確認・比較を行う機能が備わっているためである。

【0059】第2の効果は、配布するファイル容量を小さくできることにある。その理由は、配布ファイルに実行プログラム等、容量の大きなファイルが含まれず、設定情報の文字情報のみを配布するためである。

【0060】第3の効果は、既に構築されているアプリケーションに対し、新しい実行プログラムを追加することなくアプリケーション用データベースの内容を変更できることにある。その理由は、データベース更新処理装置はアプリケーションに既に含まれており、必要な情報は文字情報のみであるためである。

【0061】第4の効果は、保守が容易であり、短期間に作成、修正できることにある。その理由は、配布ファイルが文字情報であるため、実行プログラムの作成が必要無いためである。

【0062】第5の効果は、利用者の運用ミスを防げることにある。その理由は、実行時に利用者が指示する内容が、実行する設定ファイルの選択と、簡単な問合せ形式に限定できるためである。

【図面の簡単な説明】

【図1】本発明の第1実施例のシステム構成図である。

【図2】同第1実施例の全体の概略フローチャートである。

【図3】同第1実施例における定義ファイルの実行部の読出しのフローチャートである。

【図4】図3の読出しにおいて、定義ファイルの行がバージョンの確認である場合の処理である。

【図5】図3のステップB03にて読み出した定義ファイルの行がバージョンの変更である場合の処理である。

【図6】図3のステップB03にて読み出した定義ファイルの行が、条件で実行されるSQLである場合の処理である。

【図7】図3のステップB03にて読み出した定義ファイルの行が無条件に実行されるSQLである場合の処理である。

【図8】図3のステップB03にて読み出した定義ファ

11

イルの行が利用者への問合せである場合の処理である。  
 【図9】図3のステップB03にて読み出した定義ファイルの行が定義ファイル読み出し行の移動である場合の処理である。

【図10】アプリケーション用データベースと、バージョン管理部と、定義ファイルと、条件確認用SQLと、条件実行用SQLと、実行用SQLの具体例である。

【図11】図10の実行結果を示す図である。

【図12】図10の別の実行結果を示す図である。

【図13】図10の更に別の実行結果を示す図である。

【図14】本発明の第2実施例のシステム構成図である。

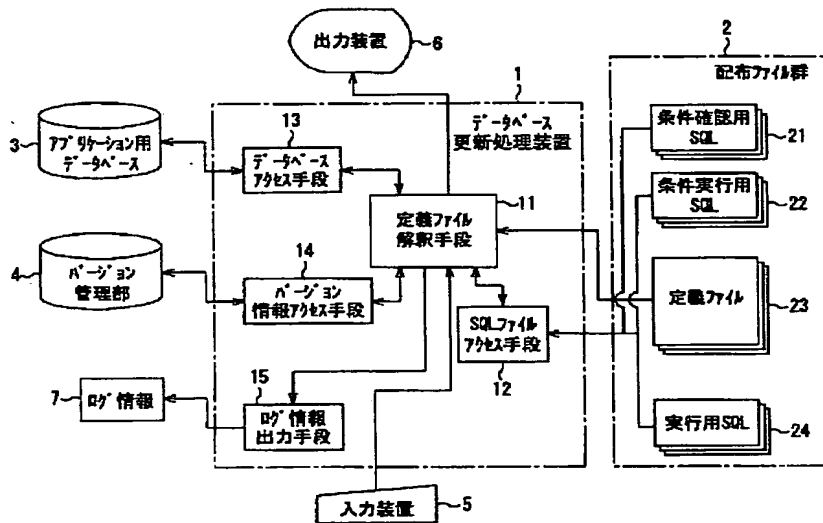
【図15】同第2実施例の動作フローチャートである。

【図16】同第2実施例の具体例を示す図である。

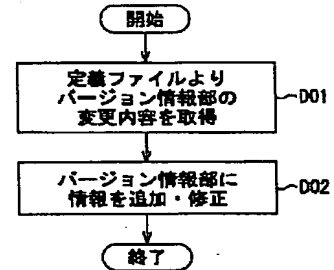
【符号の説明】

- \* 1 データベース更新処理装置  
 2 配布ファイル群  
 3 アプリケーション用データベース  
 4 バージョン管理部  
 5 入力装置  
 6 出力装置  
 11 定義ファイル解釈手段  
 12 SQLファイルアクセス手段  
 13 データベースアクセス手段  
 14 バージョン情報アクセス手段  
 15 ログ情報出力手段  
 21 条件確認用SQL  
 22 条件実行用SQL  
 23 定義ファイル  
 24 実行用SQL

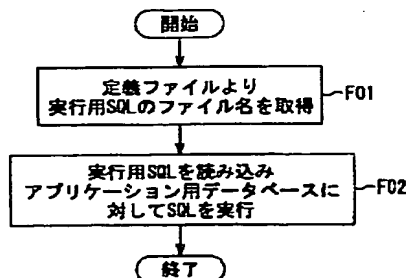
【図1】



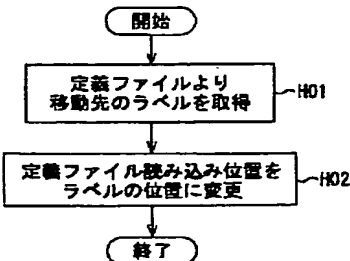
【図5】



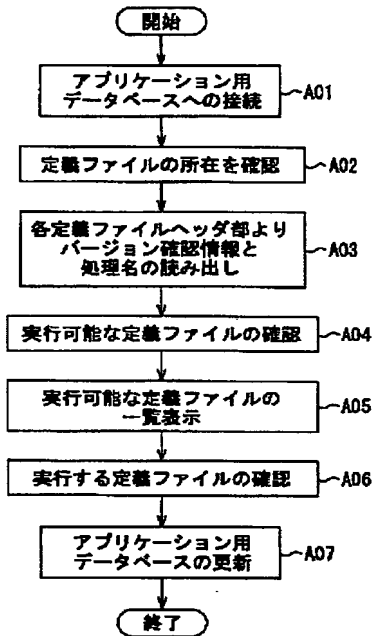
【図7】



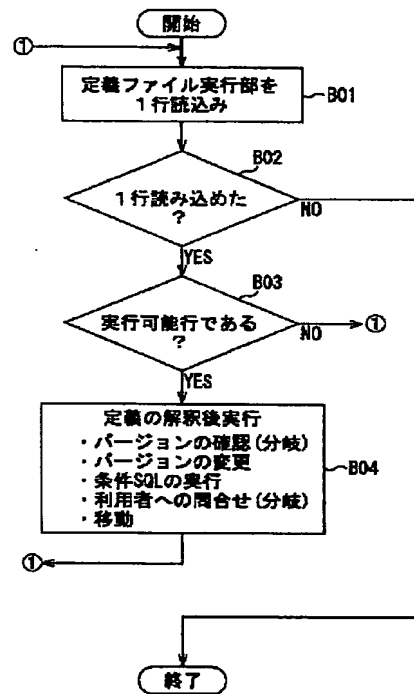
【図9】



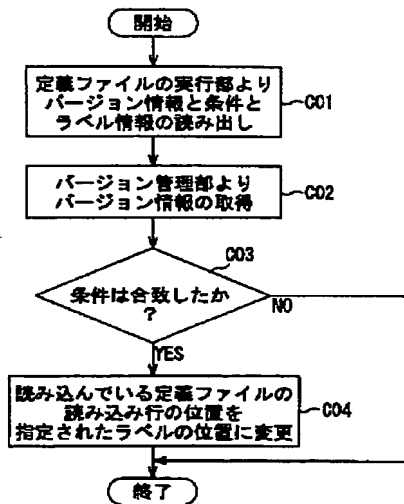
【図2】



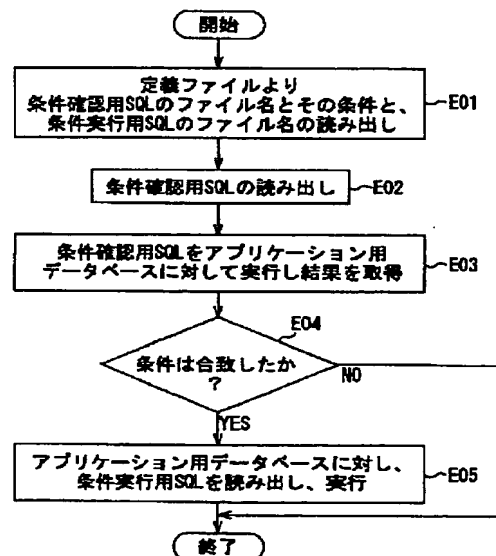
【図3】



【図4】

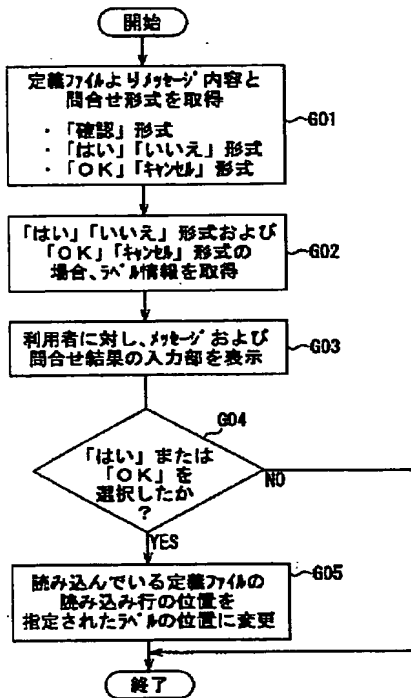


【図6】

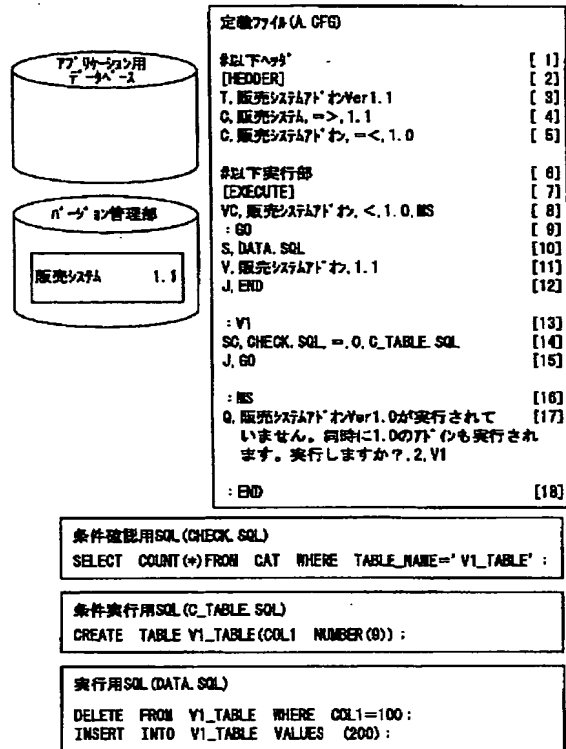




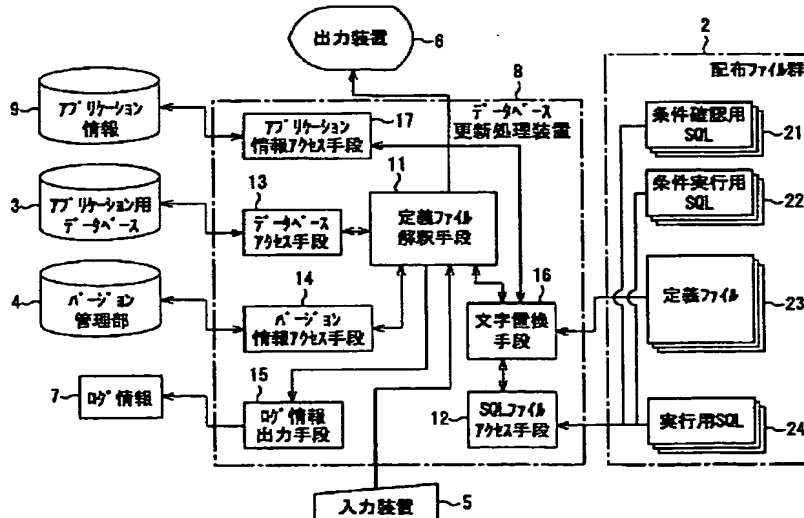
【図8】



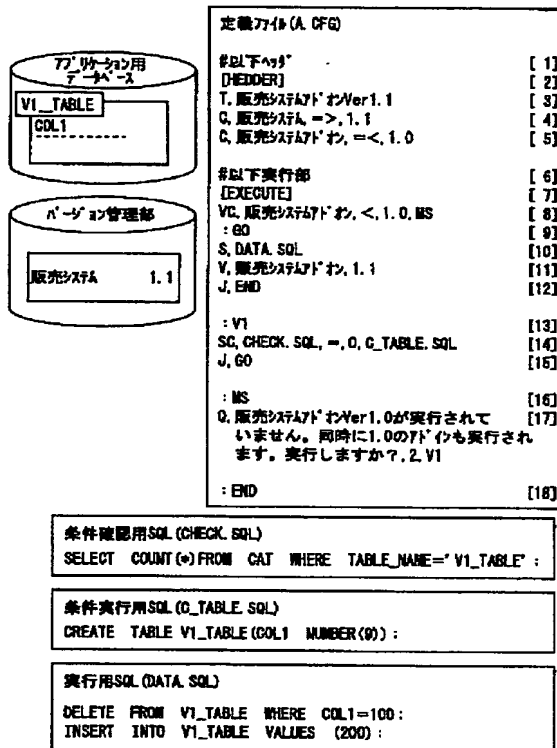
【図10】



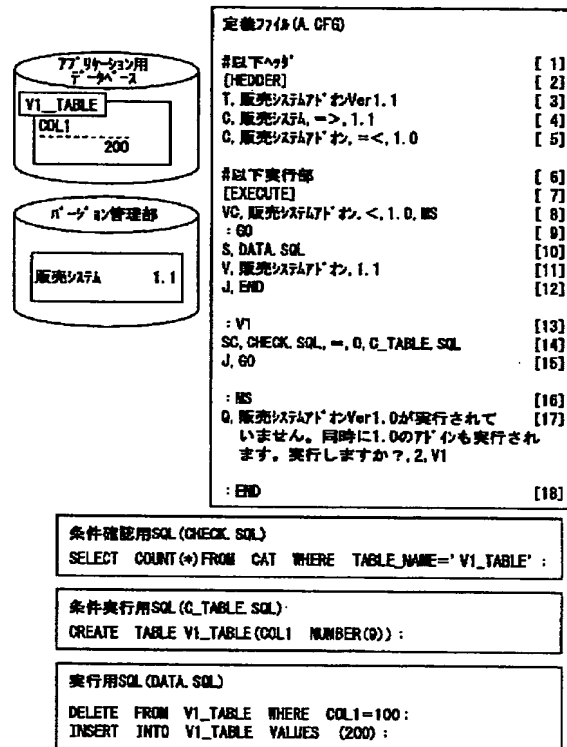
【図14】



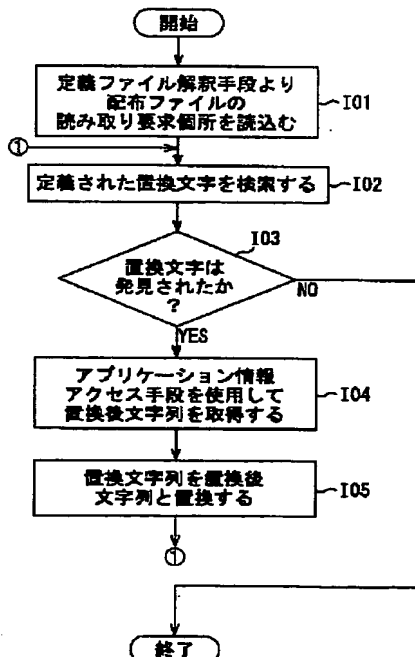
【図11】



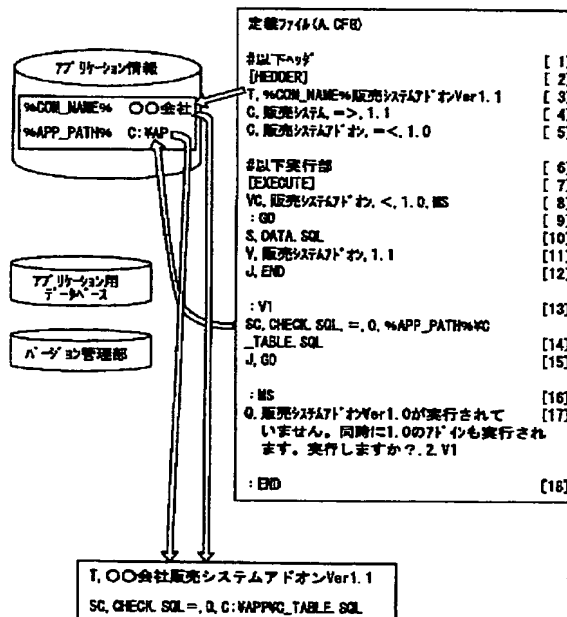
【図12】



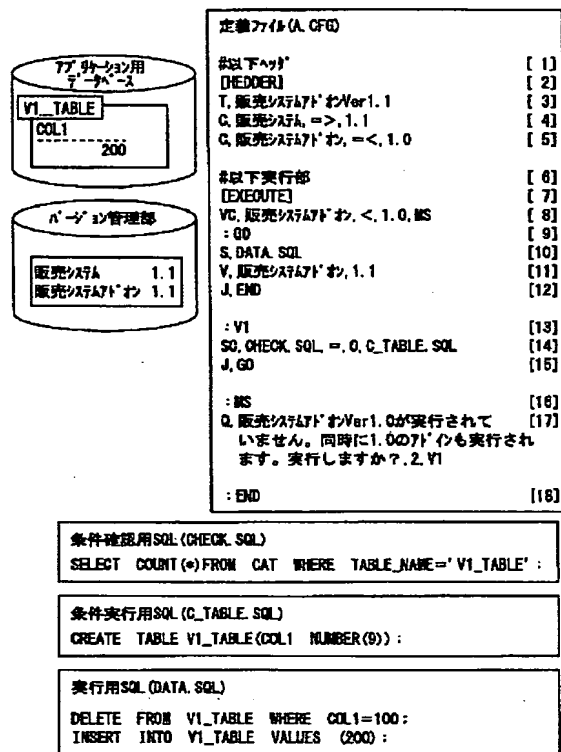
【図15】



【図16】



【図13】



**THIS PAGE BLANK (USPTO)**

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**